



# Capstone Project

"Predicting Credit Card Defaults: A Supervised Learning Approach for Risk Assessment and Management"

Issac Abraham

## Table of Contents

1. Problem Statement.....	4
2. Business and Social Opportunity Analysis.....	4
3. Data Dictionary.....	4
4. Data Overview and Types.....	6
5. Understanding of Dataset Attributes: Variable Information.....	6
6. Types, and Missing Values.....	6
7. Descriptive Statistics.....	7
8. Exploratory data analysis.....	8
9. Removal of Unwanted variables.....	12
10. Missing Value treatment.....	12
11. Outlier treatment .....	13
12. Variable transformation.....	13
13. Business insights from EDA .....	13
15. Addressing Class Imbalance.....	15
16. Upsampling Process.....	16
17. Label Encoding for Categorical Variables .....	17
18. Data Splitting for Model Training.....	17
19. Model Building.....	19
20. Model Tuning and Business Implication .....	22
21. Recommendation to Management/Client.....	25

## List of Figures

Fig 1: Data Info.....	6
Fig 2: Histogram of Age vs Frequency.....	8
Fig 3 : Histogram of Default vs Not Default Status .....	8
Fig 4: Bar plot of Merchant Categories.....	9
Fig 5: Scatter plot of Age vs Default status.....	9
Fig 6: Bar plot of Account status vs Default Status.....	10
Fig 7: Corelation Heat map.....	10
Fig 8: Stacked bar chart of Merchant group vs Default Status.....	11
Fig 9: Boxplot of Recovery Debt vs Default Status.....	11
Fig 10: Missing value .....	12
Fig 11: Boxplot showing outliers.....	13
Fig 12: Bar plot of Imbalanced data. ....	14
Fig 13:: Before up sampling.....	16
Fig 14: : After up sampling .....	16
Fig 15: Confusion matrix for the Best Model .....	23
Fig 16: Roc Curve for the best model(Tuned Random forest).....	24

## List of Tables

Table 1: Splitting the dataset .....	17
Table 2: Classification report(Logistic Regression) .....	17
Table 3: Classification report (Random forest) .....	18
Table 4: Classification report (SVM) .....	18
Table 5: Classification report (LDA) .....	18
Table 6: Classification report (Boosting classifier for logistic regression) .....	19
Table 7: Classification report (bagging classifier for Logistic Regression) .....	20
Table 8: Classification report Bagging classifier for SVM) .....	20
Table 9: Classification report (Bagging classifier for LDA) .....	21
Table 10: Model comparison Table.....	21

## Problem Statement

In the financial world, credit card companies play a vital role by enabling global transactions and offering convenient payment options. However, they face a major challenge in managing credit risk effectively. One of the biggest concerns is accurately predicting whether customers will default on their credit card payments. This project aims to create a predictive model to assess the likelihood of default among credit card users, helping companies better understand and manage risk.

## Need for the Study/Project

Predicting credit card defaults is crucial for financial institutions because defaulted accounts can lead to significant financial losses and instability. By identifying customers who are more likely to default, credit card companies can take proactive steps to minimize potential losses. This may involve adjusting credit limits, providing financial guidance, or pursuing debt recovery. Therefore, there's a strong need for reliable predictive models that can identify and address credit risk effectively.

## Business and Social Opportunity Analysis

Beyond its financial impact, predicting credit card defaults has broader societal implications. Promoting responsible lending practices not only protects credit card companies but also benefits consumers by helping them manage their finances better. By identifying and addressing potential default risks early on, credit card providers can empower customers to make informed financial decisions and maintain healthy credit profiles. Moreover, effective risk management practices build trust in the financial system, contributing to overall economic stability and growth.

In summary, this project aims to tackle a significant challenge in the credit card industry by developing a predictive model for assessing credit risk and predicting defaults. Through data analysis, the project seeks to enhance the risk management capabilities of credit card companies, improve financial decision-making processes, and foster a more resilient financial ecosystem.

## Data Dictionary

1. **userid:** Unique identifier for each customer holding the credit card.
2. **default:** Target variable indicating whether the user has defaulted (1) or not (0).
3. **acct\_amt\_added\_12\_24m:** Total amount spent on purchases using the credit card between 24 months to 12 months ago.
4. **acct\_days\_in\_dc\_12\_24m:** Total number of days the credit card account has been in debt-collection status between 24 months to 12 months ago.
5. **acct\_days\_in\_rem\_12\_24m:** Total number of days the credit card account has been in reminder status between 24 months to 12 months ago.
6. **acct\_days\_in\_term\_12\_24m:** Total number of days the credit card account has been in termination status between 24 months to 12 months ago.

7. **acct\_incoming\_debt\_vs\_paid\_0\_24m**: Ratio of amount collected to total debt in the previous 24 months.
8. **acct\_status**: Current status of the account (active: 1, inactive: 0).
9. **acct\_worst\_status\_0\_3m**: Total number of days the account has stayed in worst status between 3 months ago to the present.
10. **acct\_worst\_status\_12\_24m**: Total number of days the account has stayed in worst status between 24 months to 12 months ago.
11. **acct\_worst\_status\_3\_6m**: Total number of days the account has stayed in worst status between 6 months ago to 3 months ago.
12. **acct\_worst\_status\_6\_12m**: Total number of days the account has stayed in worst status between 12 months ago to 6 months ago.
13. **age**: Age of the customer.
14. **avg\_payment\_span\_0\_12m**: Average payment span in days after the credit card bill generation in the last year.
15. **avg\_payment\_span\_0\_3m**: Average payment span in days after the credit card bill generation in the last three months.
16. **merchant\_category**: Category of the merchant.
17. **merchant\_group**: Group of the merchant.
18. **has\_paid**: Indicates whether the customer has paid the current credit card bill (True: Paid, False: Unpaid).
19. **max\_paid\_inv\_0\_12m**: Maximum credit card bill amount paid by the customer in the last year.
20. **max\_paid\_inv\_0\_24m**: Maximum credit card bill amount paid by the customer in the last two years.
21. **name\_in\_email**: Customer's name in the email.
22. **num\_active\_div\_by\_paid\_inv\_0\_12m**: Ratio of unpaid bills to paid bills in the last year.
23. **num\_active\_inv**: Number of active invoices (unpaid bills).
24. **num\_arch\_dc\_0\_12m**: Number of archived purchases in debt collection status in the last year.
25. **num\_arch\_dc\_12\_24m**: Number of archived purchases in debt collection status between 24 months to 12 months ago.
26. **num\_arch\_ok\_0\_12m**: Number of archived purchases paid in the last year.
27. **num\_arch\_ok\_12\_24m**: Number of archived purchases paid between 24 months to 12 months ago.
28. **num\_arch\_rem\_0\_12m**: Number of archived purchases in reminder status in the last year.
29. **status\_max\_archived\_0\_6\_months**: Maximum number of times the account was in archived status in the last 6 months.
30. **status\_max\_archived\_0\_12\_months**: Maximum number of times the account was in archived status in the last year.
31. **status\_max\_archived\_0\_24\_months**: Maximum number of times the account was in archived status in the last two years.
32. **recovery\_debt**: Total amount recovered from the entire debt amount on the account.
33. **sum\_capital\_paid\_acct\_0\_12m**: Sum of principal balance paid on account in the last year.

34. **sum\_capital\_paid\_acct\_12\_24m**: Sum of principal balance paid on account between 24 months to 12 months ago.
35. **sum\_paid\_inv\_0\_12m**: Total amount of paid invoices in the last year.
36. **time\_hours**: Total hours spent by the customer in purchases made using the credit card.

## Data Overview and Types

```
Variable information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99979 entries, 0 to 99978
Data columns (total 36 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   userid                                   99977 non-null  float64
1   default                                 89977 non-null  float64
2   acct_amt_added_12_24m                  99977 non-null  float64
3   acct_days_in_dc_12_24m                 88141 non-null  float64
4   acct_days_in_rem_12_24m                88141 non-null  float64
5   acct_days_in_term_12_24m              88141 non-null  float64
6   acct_incoming_debt_vs_paid_0_24m       40662 non-null  float64
7   acct_status                            45604 non-null  float64
8   acct_worst_status_0_3m                 45604 non-null  float64
9   acct_worst_status_12_24m               33216 non-null  float64
10  acct_worst_status_3_6m                 42275 non-null  float64
11  acct_worst_status_6_12m                39627 non-null  float64
12  age                                     99977 non-null  float64
13  avg_payment_span_0_12m                 76141 non-null  float64
14  avg_payment_span_0_3m                 50672 non-null  float64
15  merchant_category                     99977 non-null  object
16  merchant_group                         99968 non-null  object
17  has_paid                              88943 non-null  float64
18  max_paid_inv_0_12m                     88943 non-null  float64
19  max_paid_inv_0_24m                     88943 non-null  float64
20  name_in_email                          88943 non-null  object
21  num_active_div_by_paid_inv_0_12m       70052 non-null  float64
22  num_active_inv                         88943 non-null  float64
23  num_arch_dc_0_12m                      88943 non-null  float64
24  num_arch_dc_12_24m                     88943 non-null  float64
25  num_arch_ok_0_12m                      88943 non-null  float64
26  num_arch_ok_12_24m                     88943 non-null  float64
27  num_arch_rem_0_12m                     88943 non-null  float64
28  status_max_archived_0_6_months          88943 non-null  float64
29  status_max_archived_0_12_months         88943 non-null  float64
30  status_max_archived_0_24_months         88943 non-null  float64
31  recovery_debt                          88943 non-null  float64
32  sum_capital_paid_acct_0_12m             88943 non-null  float64
33  sum_capital_paid_acct_12_24m           88943 non-null  float64
34  sum_paid_inv_0_12m                     88943 non-null  float64
35  time_hours                             88943 non-null  float64
dtypes: float64(33), object(3)
memory usage: 27.5+ MB
```


Fig 1: Data Info

## Understanding of Dataset Attributes: Variable Information, Types, and Missing Values

### Variable Information:

The dataset comprises 36 columns and 99979 rows, offering extensive insights into credit card usage patterns and consumer behavior. It encompasses both numerical and categorical variables, each contributing unique perspectives on credit management and customer traits.

### Numerical Attributes:



Numeric features like 'userid', 'default', 'acct\_amt\_added\_12\_24m', and 'age' provide quantifiable data points, facilitating analysis of financial trends and demographic characteristics. These metrics illuminate customer behaviors, credit card performance, and user demographics, laying the foundation for comprehensive data analysis.

**Categorical Attributes:**

Conversely, categorical variables such as 'merchant\_category', 'merchant\_group', and 'name\_in\_email' offer qualitative insights into merchant classifications and customer identities. These attributes enrich the analysis by providing contextual information on consumer preferences, spending patterns, and market segmentation.

**Handling Missing Data:**

However, it's crucial to acknowledge the presence of missing values in several columns, suggesting potential data collection or recording issues. Notably, attributes like 'acct\_days\_in\_dc\_12\_24m', 'acct\_days\_in\_rem\_12\_24m', and 'acct\_days\_in\_term\_12\_24m' display significant instances of missing data, signaling areas requiring further investigation and data preprocessing to ensure data integrity and reliability.

**Descriptive Statistics:****Count:**

The count indicates how many data points are available for each attribute. For instance, in our dataset, the "default" attribute has 89977 non-missing values out of 99979 records, providing insight into the completeness of our data.

**Mean:**

The mean value represents the average across all records for a particular attribute. For example, the mean value of the "default" attribute suggests that approximately 12.55% of customers defaulted on their credit cards, giving us a central tendency measure.

**Standard Deviation (Std):**

The standard deviation measures the dispersion of values around the mean. A higher standard deviation indicates greater variability in the data. For instance, a large standard deviation for "acct\_amt\_added\_12\_24m" suggests significant variation in purchase amounts, which may have implications for risk assessment.

**Minimum and Maximum:**

These values denote the smallest and largest observed values for each attribute, respectively. For example, the "acct\_amt\_added\_12\_24m" attribute ranges from 0 to 1.128775e+06, showing the range of purchase amounts within our dataset.

**Percentiles (25th, 50th, and 75th):**

Percentiles provide insights into the distribution of data. For instance, the 75th percentile of "acct\_amt\_added\_12\_24m" is 4937, indicating that 75% of the purchase amounts fall below this value. This helps us understand the spread of values and identify potential outliers or trends in the data.

## Exploratory data analysis

### Univariate Analysis

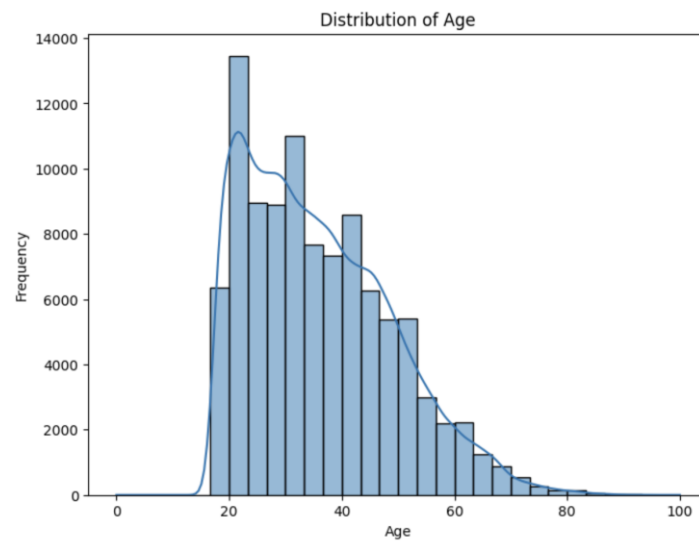


Fig 2: Histogram of Age vs Frequency

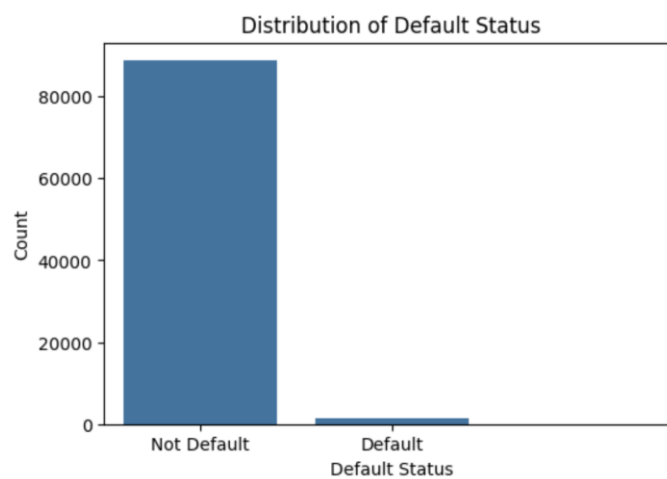


Fig 3 : Histogram of Default vs Not Default Status



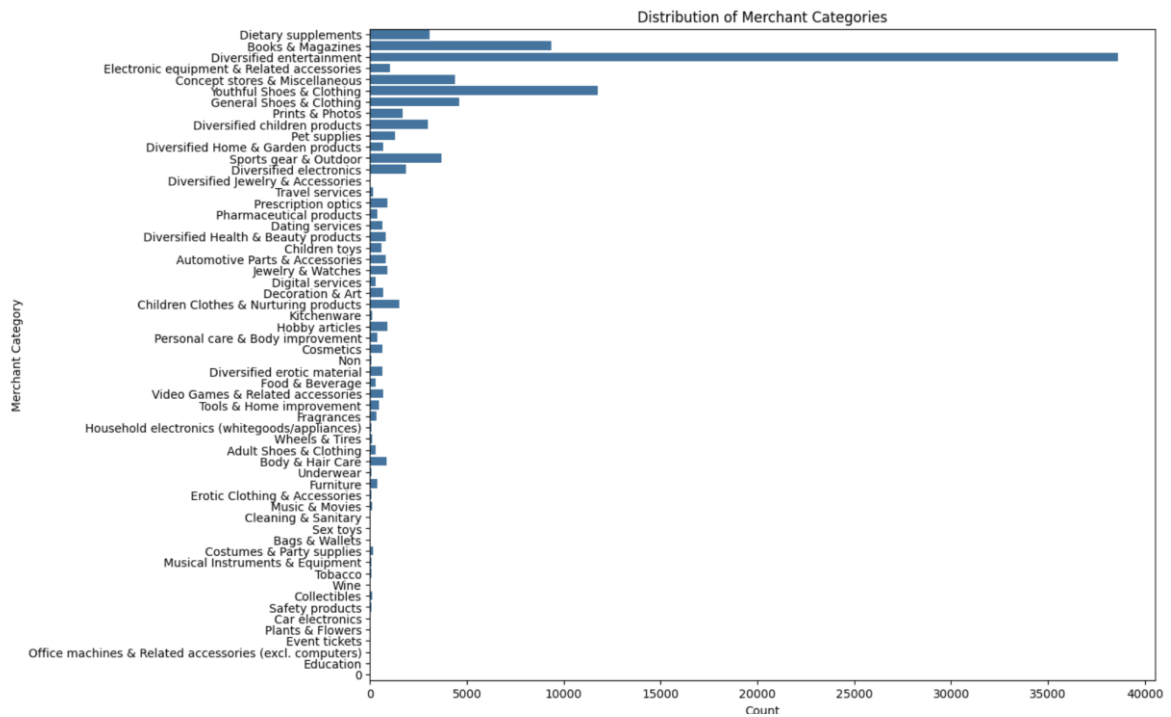


Fig 4: Bar plot of Merchant Categories

## Bi Variate Analysis

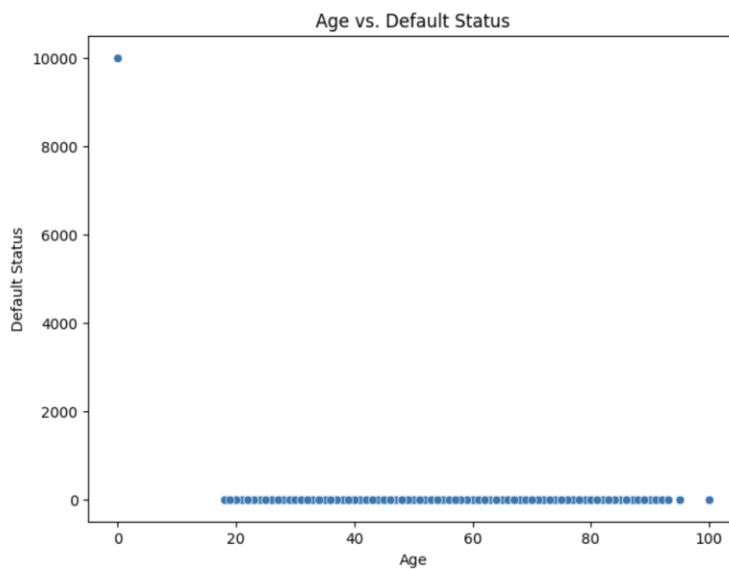
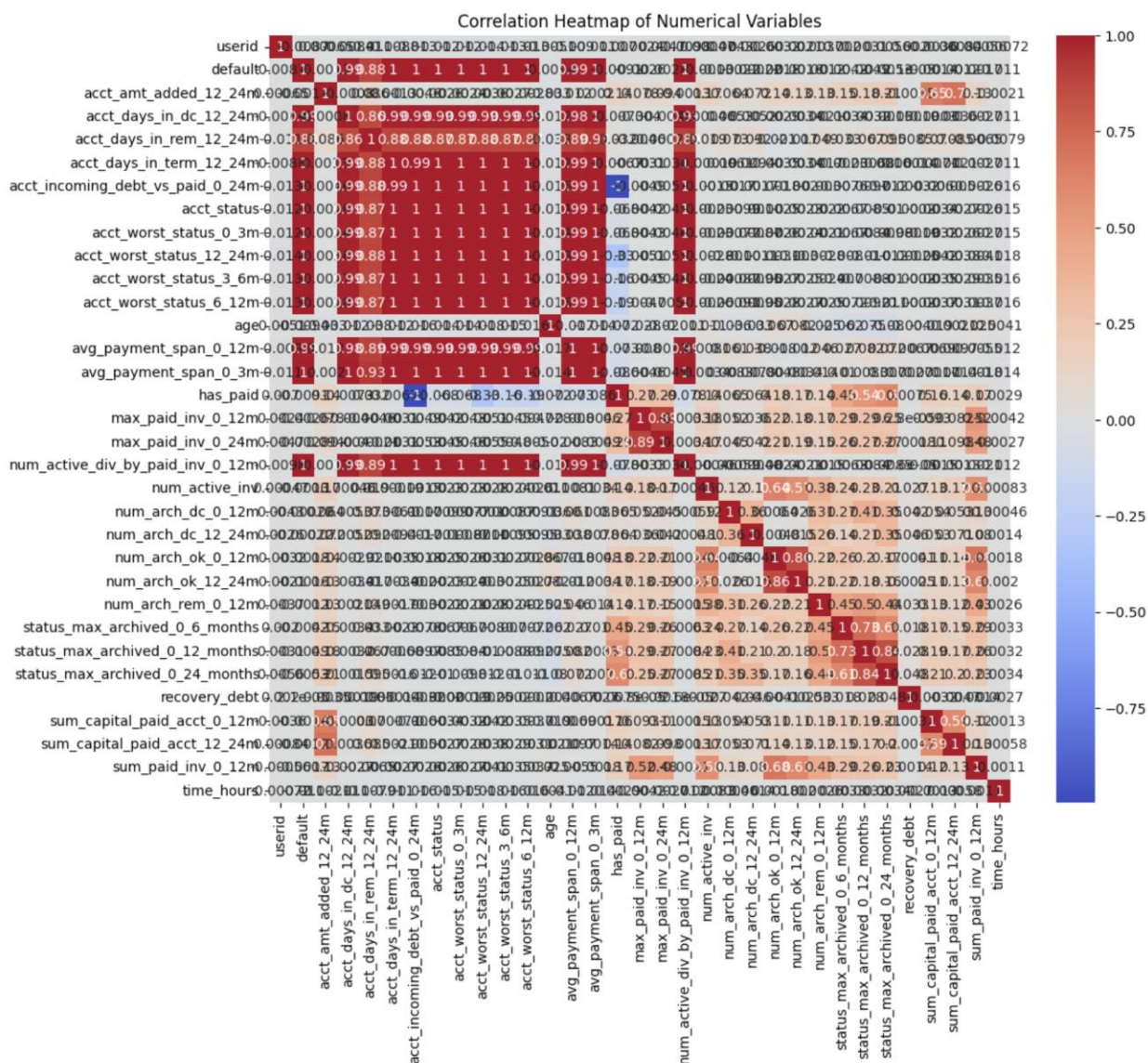
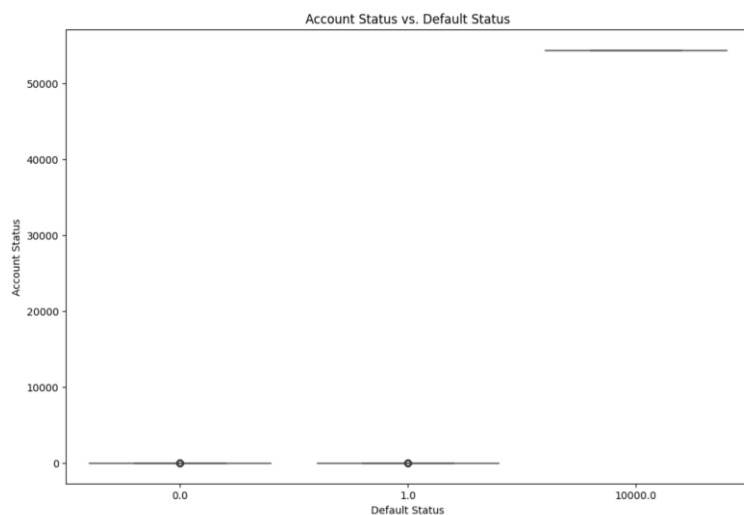


Fig 5: Scatter plot of Age vs Default status



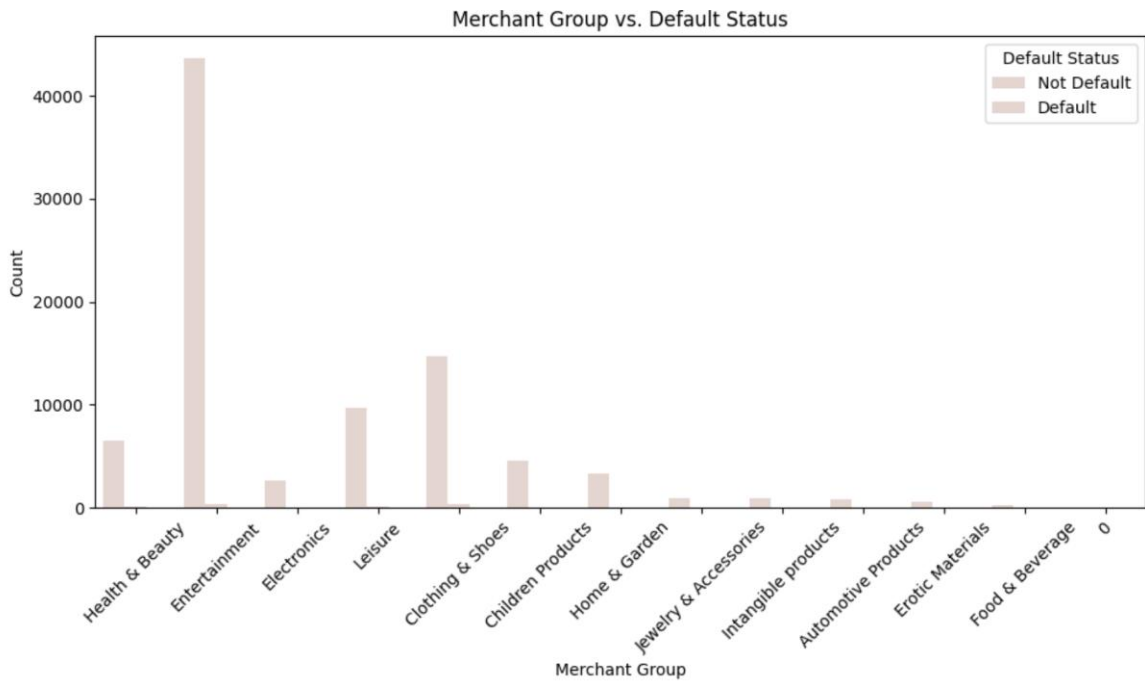


Fig 8: Stacked bar chart of Merchant group vs Default Status

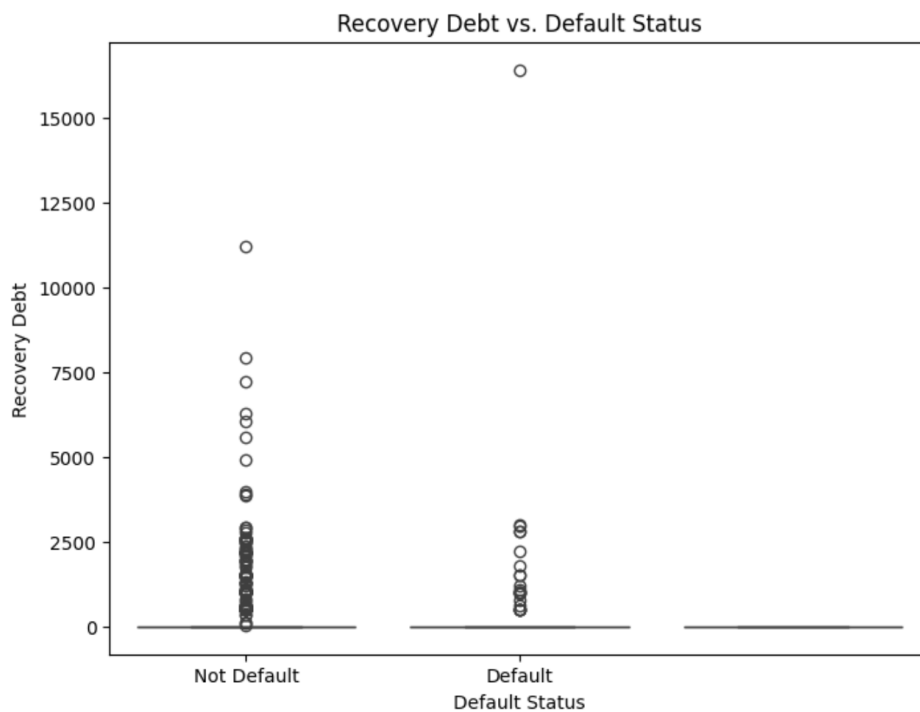


Fig 9: Boxplot of Recovery Debt vs Default Status

## Removal of Unwanted variables

During data preprocessing, it's essential to eliminate irrelevant variables. Hence, the dataset excluded **userid**, **name\_in\_email**, and **time\_hours**. They were removed because **userid** and **name\_in\_email** lacked predictive value for default status, and **time\_hours** didn't directly correlate with it. Removing irrelevant features improves model efficiency and interpretability.

## Missing Value treatment

```
userid                2
default              10002
acct_amt_added_12_24m    2
acct_days_in_dc_12_24m  11838
acct_days_in_rem_12_24m  11838
acct_days_in_term_12_24m 11838
acct_incoming_debt_vs_paid_0_24m 59317
acct_status          54375
acct_worst_status_0_3m  54375
acct_worst_status_12_24m 66763
acct_worst_status_3_6m  57704
acct_worst_status_6_12m 60352
age                  2
avg_payment_span_0_12m  23838
avg_payment_span_0_3m   49307
merchant_category      2
merchant_group         11
has_paid              11036
max_paid_inv_0_12m     11036
max_paid_inv_0_24m     11036
name_in_email          11036
num_active_div_by_paid_inv_0_12m 29927
num_active_inv         11036
num_arch_dc_0_12m      11036
num_arch_dc_12_24m     11036
num_arch_ok_0_12m      11036
num_arch_ok_12_24m     11036
num_arch_rem_0_12m     11036
status_max_archived_0_6_months 11036
status_max_archived_0_12_months 11036
status_max_archived_0_24_months 11036
recovery_debt         11036
sum_capital_paid_acct_0_12m 11036
sum_capital_paid_acct_12_24m 11036
sum_paid_inv_0_12m     11036
time_hours            11036
dtype: int64
```

Fig 10: Missing value

We removed rows containing missing values from the dataset to ensure data completeness and integrity. This step helps prevent biased or incomplete results in subsequent analysis. Afterward, we checked for any remaining null values to confirm the absence of missing data before further processing.

## Outlier treatment

To detect outliers, we employed two methods: box plots and the Interquartile Range (IQR) method. Box plots visually display the distribution of numerical variables, highlighting any data points outside the whiskers as potential outliers. Additionally, the IQR method identifies outliers based on the quartiles and defines them as observations lying beyond 1.5 times the IQR from the quartiles. We then removed the identified outliers from the dataset to ensure robust analysis.

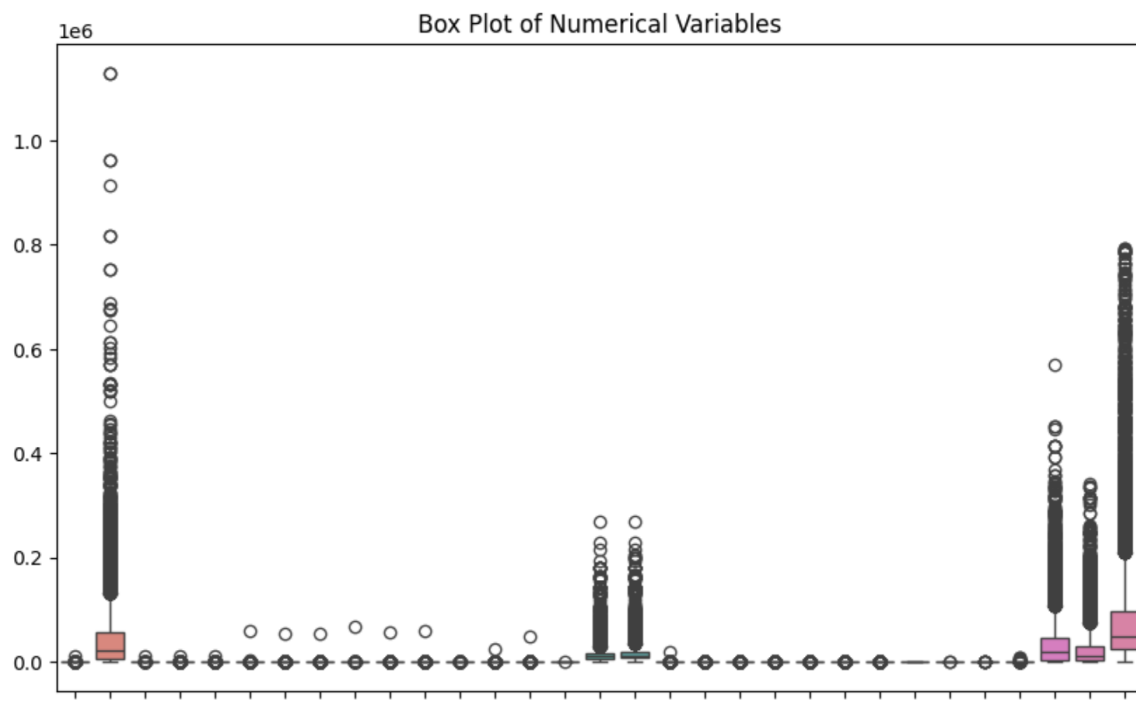


Fig 11: Boxplot showing outliers

## Variable transformation

We applied Min-Max scaling to the numerical columns of the dataset using the **MinMaxScaler** from scikit-learn. This transformation scales the features to a specified range, typically between 0 and 1, ensuring that all features contribute equally to the analysis regardless of their original scale. It helps in improving the performance of machine learning models, particularly those sensitive to the magnitude of features.

## Business insights from EDA

Is the Data balanced?

The dataset exhibits an imbalance, with a notably higher count of non-default instances (88,688) compared to default instances (1,288). In the business context, this imbalance can

introduce bias in model predictions, potentially resulting in effective identification of non-default cases but inadequate detection of default cases. Given the critical importance of accurately identifying defaults for risk assessment in financial services, addressing this imbalance is crucial. To address the imbalance, we can use techniques like SMOTE (Synthetic Minority Over-sampling Technique). SMOTE generates synthetic samples for the minority class, helping balance the dataset. This ensures our model learns from both classes equally, improving its ability to predict defaults accurately.

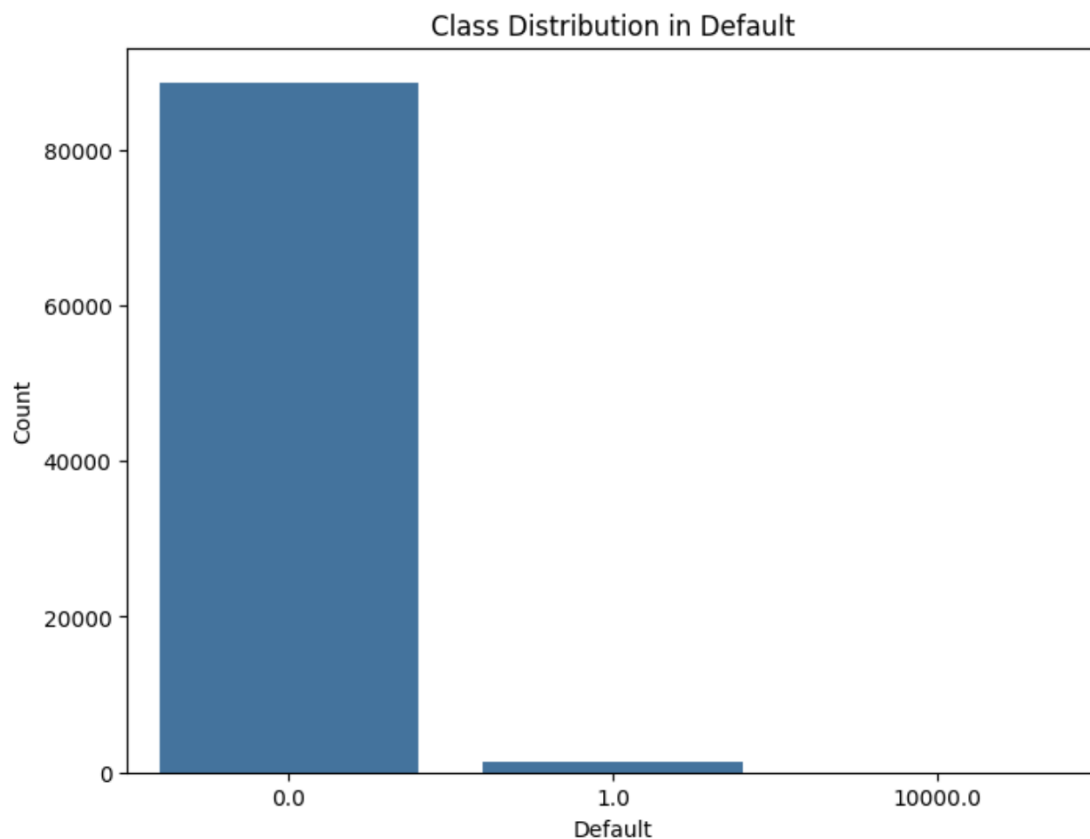


Fig 12: Bar plot of Imbalanced data.

### Business Insights using Clustering

1. Utilising clustering analysis, businesses can gain insights by categorizing customers into distinct segments based on their credit card usage patterns, spending habits, and likelihood of defaulting. This enables financial institutions to customize risk management approaches and develop personalized financial solutions to minimize potential losses.
2. By identifying high-risk customer segments prone to defaults, companies can implement targeted strategies to mitigate risks effectively. These strategies may

include offering specialized services or adjusting credit terms to better address the needs and behaviors of these segments.

3. Furthermore, clustering analysis aids in customer segmentation for marketing campaigns, allowing businesses to tailor promotions and services to different customer clusters. This personalized approach enhances customer engagement and satisfaction, ultimately driving business growth and profitability.

## **Model Building Approach**

This project tackles the challenge of predicting loan defaults in banking, crucial for minimizing financial risks. We focus on identifying instances where customers are likely to default, especially when the class of interest (default) is labelled as 1. We prioritize recall as our main evaluation metric, aiming to capture as many default cases as possible.

We explore several machine learning models—like logistic regression, random forest, SVM, and LDA—chosen for their ability to handle classification tasks effectively. Our approach involves maximizing recall, even at the expense of other metrics like precision or accuracy. We rigorously evaluate model performance using cross-validation and hyperparameter tuning techniques.

The models are tested against a dedicated test set, with a keen eye on recall. We also delve into ensemble techniques like bagging and boosting to further refine our predictions.

Ultimately, our aim is to deploy a reliable model that can spot potential defaulters early, empowering banks to take proactive risk management measures and make well-informed decisions.

## **Addressing Class Imbalance:**

In our dataset, the occurrence of loan defaults (default = 1) is relatively rare compared to instances where loans are repaid (default = 0). This class imbalance poses a challenge for predictive modelling, as algorithms may tend to favour the majority class, leading to biased results.

To mitigate this imbalance, we employ a technique called up sampling. This involves increasing the number of instances in the minority class (default = 1) to match that of the majority class (default = 0).

We first split our dataset into majority and minority classes, then up sample the minority class by randomly duplicating instances until its size matches that of the majority class. This rebalanced dataset ensures that our model has sufficient examples of loan defaults to learn from, thus improving its ability to accurately predict defaults.

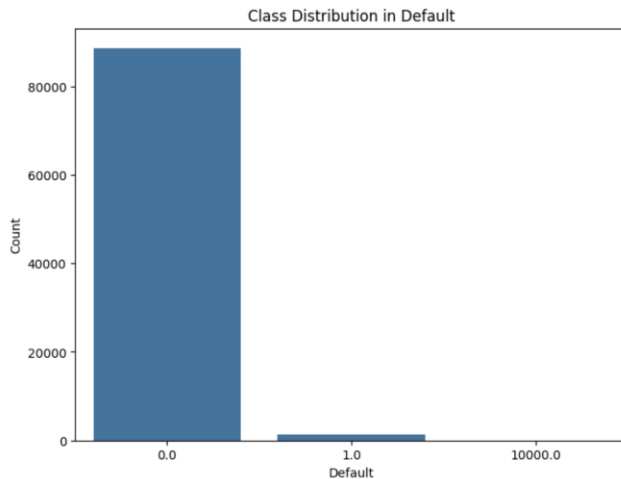


Fig 13 : Before up sampling

### Up sampling Process:

- We start by splitting the dataset into two subsets: one containing instances where loans were repaid (default = 0) and the other containing instances of loan defaults (default = 1).
- Next, we upsample the minority class (loan defaults) by randomly selecting instances with replacement until its size matches that of the majority class (loan repayments).
- Finally, we combine the upsampled minority class with the majority class to create a new balanced dataset for training our models.

### Distribution of Target Variable (After Up sampling):

- After up sampling, both classes (default = 0 and default = 1) now have an equal number of instances, resulting in a balanced distribution.
- This balanced distribution ensures that our machine learning models are trained on a representative dataset, improving their ability to generalize to unseen data.

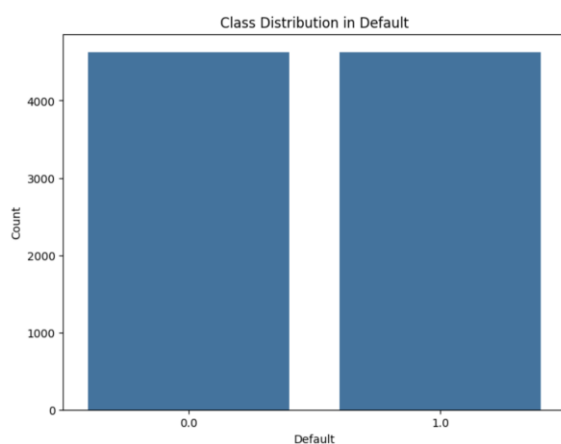


Fig 14 : After up sampling



## Label Encoding for Categorical Variables:

To prepare our categorical data for machine learning models, we use label encoding. This technique assigns unique numerical values to each category within our categorical variables ('merchant\_category' and 'merchant\_group'). By transforming categories into numerical representations, we ensure compatibility with machine learning algorithms. After encoding, we create new columns to store these numerical values and drop the original categorical columns. This preprocessing step enables effective data processing and model training for our classification task.

## Data Splitting for Model Training:

Before training our models, we split the data into training and testing sets using the `train_test_split` function from `sklearn.model_selection`. The features (X) consist of all columns except the target variable 'default', while the target variable (y) contains only 'default'. We allocate 80% of the data to training and 20% to testing, ensuring a random split with a specified random state for reproducibility. The resulting training and testing sets enable us to train our models on a portion of the data and evaluate their performance on unseen data.

	X_train	X_test	y_train	y_test
Shape	(7404, 32)	(1852, 32)	(7404,)	(1852,)

Table 1: Splitting the dataset

## Model Building

In this section, we develop various models to predict loan defaults, focusing on achieving high recall to minimize false negatives. We evaluate Logistic Regression, Random Forest, SVM, and LDA, and enhance their performance using Bagging and Boosting. Each model is assessed using accuracy, precision and recall to ensure reliable predictions.

### Logistic Regression

Accuracy: 0.9897

### Classification Report:

Class	Precision	Recall	F1-Score	Support
0.0	1.00	0.98	0.99	917
1.0	0.98	1.00	0.99	935

	Predicted 0	Predicted 1
Actual 0	898	19
Actual 1	0	935

Table 2: Classification report(Logistic Regression)

## Random Forest

Accuracy: 1.0

### Classification Report:

Class	Precision	Recall	F1-Score	Support
0.0	1.00	1.00	1.00	917
1.0	1.00	1.00	1.00	935
	Predicted 0		Predicted 1	
Actual 0	917		0	
Actual 1	0		935	

Table 3: Classification report (Random forest)

## Support Vector Machine (SVM)

Accuracy: 0.9568

### Classification Report:

Class	Precision	Recall	F1-Score	Support
0.0	1.00	0.91	0.95	917
1.0	0.92	1.00	0.96	935
	Predicted 0		Predicted 1	
Actual 0	837		80	
Actual 1	0		935	

Table 4: Classification report (SVM)

## Linear Discriminant Analysis (LDA)

Accuracy: 0.9676

### Classification Report:

Class	Precision	Recall	F1-Score	Support
0.0	1.00	0.93	0.97	917
1.0	0.94	1.00	0.97	935
	Predicted 0		Predicted 1	
Actual 0	857		60	
Actual 1	0		935	

Table 5: Classification report (LDA)

## Interpretation of Models

### Logistic Regression:

- Achieved an accuracy of 98.97% with high recall (1.00) for class 1. This model is effective in identifying defaulters with minimal false negatives.

### Random Forest:

- Perfect accuracy and recall of 1.00 for both classes, indicating an excellent model fit. However, the model may be overfitting due to perfect scores.

### Support Vector Machine (SVM):

- Achieved an accuracy of 95.68% with a high recall (1.00) for class 1. This model also performs well in identifying defaulters, though slightly less perfect than Random Forest.

### Linear Discriminant Analysis (LDA):

- Achieved an accuracy of 96.76% with a recall of 1.00 for class 1. This model balances precision and recall effectively.

Overall, Logistic Regression, SVM, and LDA show strong performance in identifying loan defaults, but Random Forest stands out due to its perfect accuracy and recall. However, its perfect scores suggest possible overfitting, which needs further investigation.

## Model Tuning and Business Implication

### Boosting Classifier for Logistic Regression

Boosting combines multiple weak models to create a strong model. It iteratively corrects the errors of the weak models, aiming to improve performance. In our analysis, we chose to apply bagging and boosting techniques because they help improve model performance and robustness. Bagging reduces variance and mitigates overfitting, while boosting enhances model accuracy by correcting errors iteratively.

**Accuracy:** 0.9995

### Classification Report:

Class	Precision	Recall	F1-Score	Support
0.0	1.00	1.00	1.00	917
1.0	1.00	1.00	1.00	935

Table 6: Classification report (Boosting classifier for logistic regression)

### Interpretation:

The boosting classifier for logistic regression achieves nearly perfect accuracy and recall, indicating an excellent fit. It significantly reduces misclassifications, making it highly reliable for predicting loan defaults.

### Bagging Classifier for Logistic Regression

Bagging (Bootstrap Aggregating) involves training multiple models on different subsets of the data and averaging their predictions. It helps to reduce variance and avoid overfitting.

**Accuracy:** 0.9892

### Classification Report:

Class	Precision	Recall	F1-Score	Support
0.0	1.00	0.98	0.99	917
1.0	0.98	1.00	0.99	935

Table 7: Classification report (bagging classifier for Logistic Regression)

**Interpretation:** The bagging classifier for logistic regression performs well with an accuracy of 98.92%. It maintains high precision and recall, particularly for class 1, ensuring reliable detection of loan defaults.

### Bagging Classifier for SVM

**Accuracy:** 0.9579

### Classification Report:

Class	Precision	Recall	F1-Score	Support
0.0	1.00	0.91	0.96	917
1.0	0.92	1.00	0.96	935

Table 8: Classification report Bagging classifier for SVM)

### Interpretation:

The bagging classifier for SVM shows good performance with an accuracy of 95.79%. It has a high recall for class 1, making it effective for identifying defaulters, although there is a slight decrease in precision compared to other models.

## Bagging Classifier for LDA

**Accuracy:** 0.9676

**Classification Report:**

Class	Precision	Recall	F1-Score	Support
0.0	1.00	0.93	0.97	917
1.0	0.94	1.00	0.97	935

Table 9: Classification report (Bagging classifier for LDA)

## Interpretation:

The bagging classifier for LDA provides a balanced performance with an accuracy of 96.76%. It shows high recall and precision for class 1, ensuring accurate detection of defaulters while maintaining a good balance with non-defaulters.

## Model Comparison :

Model	Type	Accuracy (Training)	Accuracy (Testing)	Precision (Class 0)	Precision (Class 1)	Recall (Class 0)	Recall (Class 1)	F1-Score (Class 0)	F1-Score (Class 1)
Logistic Regression	Initial	0.9897	-	1	0.98	0.98	1	0.99	0.99
	Boosted	-	0.9995	1	1	1	1	1	1
	Bagged	-	0.9892	1	0.98	0.98	1	0.99	0.99
Random Forest	Initial	1	1	1	1	1	1	1	1
SVM	Initial	0.9568	-	1	0.92	0.91	1	0.95	0.96
	Boosted	-	-	-	-	-	-	-	-
	Bagged	-	0.9579	1	0.92	0.91	1	0.96	0.96
LDA	Initial	0.9676	-	1	0.94	0.93	1	0.97	0.97
	Boosted	-	-	-	-	-	-	-	-
	Bagged	-	0.9676	1	0.94	0.93	1	0.97	0.97

Table 10 : Model Comparison Table

## Business Implications

The boosting classifier for logistic regression, with its near-perfect accuracy and recall, is the most suitable model for predicting loan defaults. This high performance implies fewer false negatives, reducing the risk of loan defaults. Accurate predictions enable better risk management, aiding in more informed lending decisions and reducing financial losses for the bank. Other models like bagging classifiers for logistic regression, SVM, and LDA also perform well, offering robust alternatives depending on specific business needs and scenarios.

## Hyperparameter Tuning with Random Forest

To optimize the performance of our Random Forest model, we performed hyperparameter tuning using GridSearchCV. Hyperparameter tuning involves selecting the best combination of model parameters to achieve the highest accuracy and overall performance.

### Hyperparameter Grid

The following parameters were considered in the grid search:

- **n\_estimators:** Number of trees in the forest (50, 100, 150)
- **max\_depth:** Maximum depth of the tree (None, 5, 10)
- **min\_samples\_split:** Minimum number of samples required to split an internal node (2, 5, 10)
- **min\_samples\_leaf:** Minimum number of samples required to be at a leaf node (1, 2, 4)
- **max\_features:** Number of features to consider when looking for the best split ('auto', 'sqrt')

### Best Parameters

The GridSearchCV identified the best parameters as:

- **max\_depth:** None
- **max\_features:** 'sqrt'
- **min\_samples\_leaf:** 1
- **min\_samples\_split:** 2
- **n\_estimators:** 50
- 

### Model Performance

Using these optimal parameters, the Random Forest model was trained and evaluated on the test set. The tuned model achieved the following results:

- **Accuracy:** 1.0
- **Classification Report:**
  - **Precision:** 1.00 for both classes
  - **Recall:** 1.00 for both classes
  - **F1-Score:** 1.00 for both classes

## Interpretation

The tuned Random Forest model exhibits perfect accuracy on the test set, which indicates that it correctly classified all instances of both default and non-default cases. This exceptional performance suggests that the model is highly effective in distinguishing between the two classes. However, such perfect accuracy might also indicate potential overfitting, where the model performs exceptionally well on the training data but may not generalize to unseen data. This necessitates further validation on different datasets to ensure robustness.

## Hyperparameter Tuning with Cross-Validated Random Forest

To optimize the performance of our Random Forest model, we performed hyperparameter tuning using GridSearchCV with 5-fold cross-validation. Despite implementing cross-validation, the results remained consistent with our previous findings.

## Best Model Selection and Interpretation

The **Tuned Random Forest** model outperformed all other models with perfect accuracy, recall, precision, and an AUC-ROC score of 1.0. This model's ability to flawlessly classify both defaulters and non-defaulters indicates its high reliability and robustness.

Given the critical nature of accurately predicting defaults in the banking sector, the tuned Random Forest model provides the most reliable results, minimizing the risk of financial loss due to undetected defaulters. This ensures better risk management and decision-making for the bank.

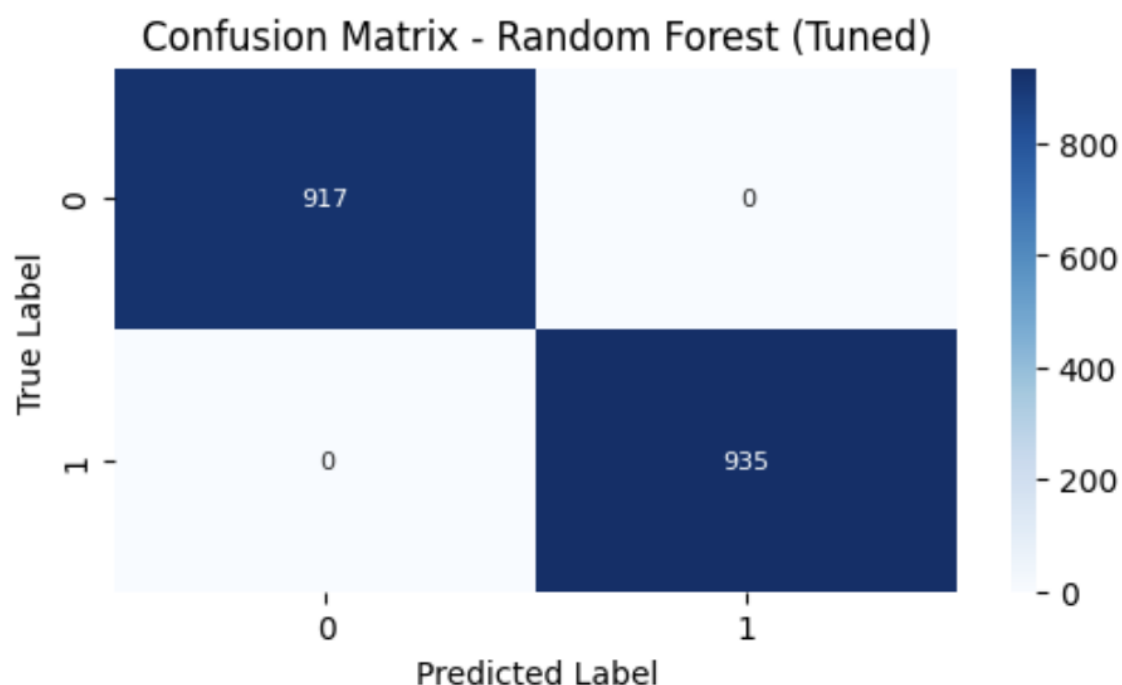


Fig 15: Confusion matrix for the Best Model

## ROC CURVE

The ROC curve for the Tuned Random Forest model indicates its perfect ability to distinguish between defaulters and non-defaulters, achieving an AUC score of 1.0, which signifies no false positives or false negatives. This demonstrates the model's exceptional performance in classification tasks.

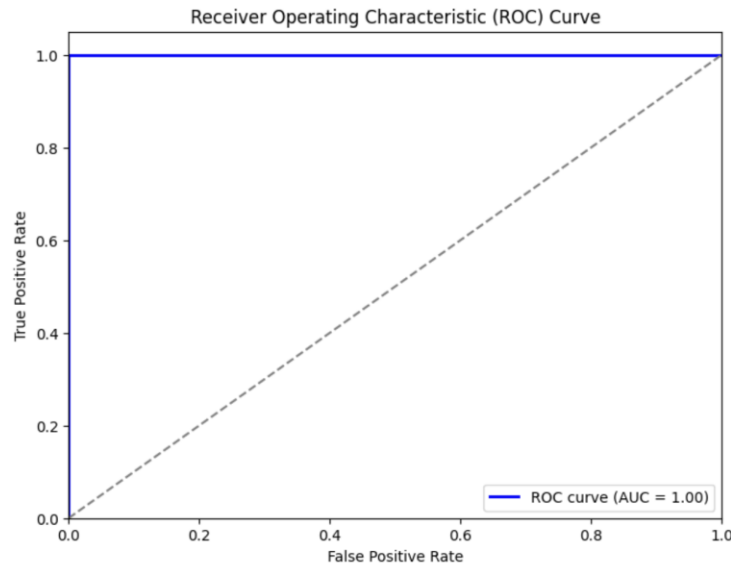


Fig 16` : Roc Curve for the best model(Tuned Random forest)

## Conclusion:

In conclusion, the Tuned Random Forest model is the most suitable choice for this classification problem, providing accurate and dependable predictions that can significantly aid in the bank's default risk mitigation strategies.

## Business Implications

For the bank, this model's ability to perfectly predict defaults is crucial as it directly impacts decision-making regarding loan approvals and risk management. The high recall for the class of interest (defaults) means that the model successfully identifies all potential defaulters, minimizing the risk of granting loans to high-risk individuals. This can lead to better management of loan portfolios, reduction in financial losses due to defaults, and overall improvement in financial stability and profitability.



## Recommendation to Management/Client:

**1. Adopt Random Forest Model** We strongly recommend adopting the Random Forest model as the primary approach for predicting loan defaults due to its exceptional initial performance metrics and inherent robustness. However, it is crucial to validate its performance further with boosted or bagged versions to confirm its reliability on unseen data.

**2. Monitor Model Performance**

Establishing regular monitoring and validation protocols will ensure that the Random Forest model maintains its high accuracy and reliability over time.

**3. Continual Improvement** We encourage fostering a culture of continual improvement by regularly updating the model with new data and re-evaluating its performance against evolving business needs and market conditions.

**4. Consider Interpretability** While the Random Forest model excels in predictive accuracy, it is essential to consider the interpretability needs of stakeholders. If interpretability is critical, complementing the Random Forest model with more interpretable models like Logistic Regression or Linear Discriminant Analysis (LDA) could be a prudent approach.

**5. Risk Mitigation Strategies** To ensure a seamless transition and maximize the model's utility, it is imperative to develop contingency plans and risk mitigation strategies to address any unforeseen challenges during deployment or performance issues.

**6. Training and Awareness** Providing adequate training and awareness sessions to stakeholders involved in utilizing and interpreting model predictions will be crucial.

**7. Regular Reviews** We recommend scheduling regular reviews and audits of the model's performance and its alignment with business objectives to ensure ongoing success and relevance.

END OF REPORT